



09/890211
PCT/FR/08983231

REC'D 05 JAN 2001
WIPO PCT

BREVET D'INVENTION

EU

CERTIFICAT D'UTILITÉ - CERTIFICAT D'ADDITION

COPIE OFFICIELLE

Le Directeur général de l'Institut national de la propriété industrielle certifie que le document ci-annexé est la copie certifiée conforme d'une demande de titre de propriété industrielle déposée à l'Institut.

08 DEC. 2000

Fait à Paris, le

Pour le Directeur général de l'Institut
national de la propriété industrielle
Le Chef du Département des brevets

A handwritten signature in black ink, appearing to read 'Martine Planche'.

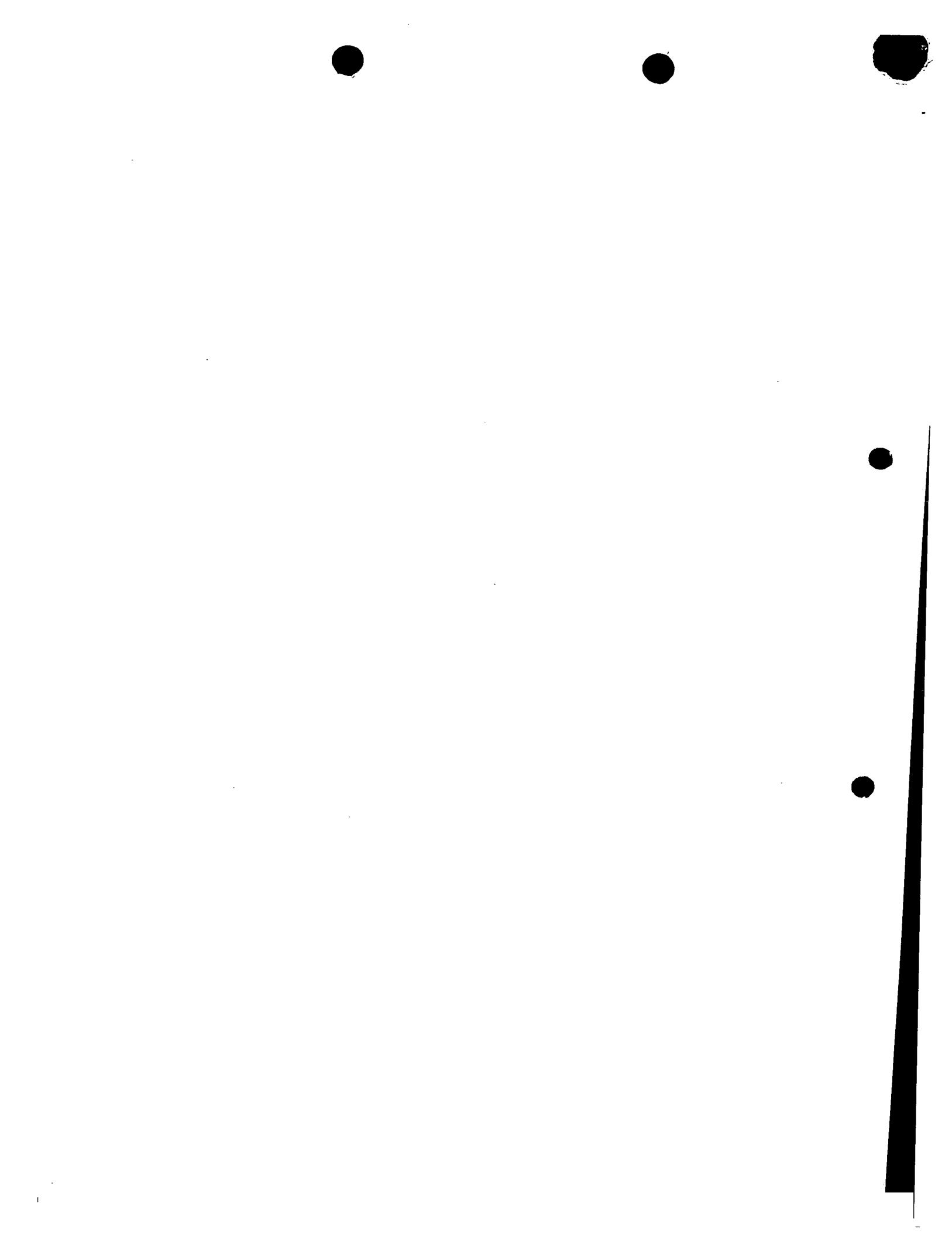
Martine PLANCHE

**PRIORITY
DOCUMENT**
SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

INSTITUT
NATIONAL DE
LA PROPRIETE
INDUSTRIELLE

SIEGE

26 bis, rue de Saint Petersbourg
75800 PARIS cedex 08
Téléphone : 01 53 04 53 04
Télécopie : 01 42 93 59 30
<http://www.inpi.fr>





**INSTITUT
NATIONAL DE
LA PROPRIÉTÉ
INDUSTRIELLE**

25 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

BREVET D'INVENTION

CERTIFICAT D'UTILITÉ

Code de la propriété intellectuelle - Livre VI

cerfa
N° 11354*01

REQUÊTE EN DÉLIVRANCE 1/2

Cet imprimé est à remplir lisiblement à l'encre noire

DB 540 W /260899

REMISE DES PIÈCES		Réervé à l'INPI
DATE	26 NOV 1999	
LIEU	75 INPI PARIS	
N° D'ENREGISTREMENT	9914882	
NATIONAL ATTRIBUÉ PAR L'INPI		
DATE DE DÉPÔT ATTRIBUÉE PAR L'INPI	26 NOV. 1999	
Vos références pour ce dossier (facultatif) FR 3863 PG		

1 NOM ET ADRESSE DU DEMANDEUR OU DU MANDATAIRE
À QUI LA CORRESPONDANCE DOIT ÊTRE ADRESSÉE
BULL S.A.
Monsieur Patrice GUILLERM
68, route de Versailles
PC : 58D20
78434 LOUVECIENNES Cedex

Confirmation d'un dépôt par télécopie		<input type="checkbox"/> N° attribué par l'INPI à la télécopie
2 NATURE DE LA DEMANDE		
Demande de brevet	<input checked="" type="checkbox"/>	
Demande de certificat d'utilité	<input type="checkbox"/>	
Demande divisionnaire	<input type="checkbox"/>	
<i>Demande de brevet initiale</i>	N°	Date / /
<i>ou demande de certificat d'utilité initiale</i>	N°	Date / /
Transformation d'une demande de brevet européen <i>Demande de brevet initiale</i>	<input type="checkbox"/>	Date / /
N°		

3 TITRE DE L'INVENTION (200 caractères ou espaces maximum)

Procédé de création de fichiers de configuration d'objets inclus dans un système informatique.

4 DÉCLARATION DE PRIORITÉ OU REQUÊTE DU BÉNÉFICE DE LA DATE DE DÉPÔT D'UNE DEMANDE ANTÉRIEURE FRANÇAISE		Pays ou organisation Date / / N° Pays ou organisation Date / / N° Pays ou organisation Date / / N° <input type="checkbox"/> S'il y a d'autres priorités, cochez la case et utilisez l'imprimé «Suite»
5 DEMANDEUR		<input type="checkbox"/> S'il y a d'autres demandeurs, cochez la case et utilisez l'imprimé «Suite»
Nom ou dénomination sociale		BULL S.A.
Prénoms		
Forme juridique		Société Anonyme
N° SIREN		6 4 2 0 5 8 7 3 9
Code APE-NAF		3 0 0 C
Adresse	68, route de Versailles	
	Rue	
Code postal et ville		78430 LOUVECIENNES
Pays		France
Nationalité		Française
N° de téléphone (facultatif)		01.39.66.67.28
N° de télécopie (facultatif)		01.39.66.71.73
Adresse électronique (facultatif)		patrice.guillerm@bull.net

REMISE DES PIÈCES	Réervé à l'INPI
DATE	26 NOV 1999
LIEU	75 INPI PARIS
N° D'ENREGISTREMENT	
NATIONAL ATTRIBUÉ PAR L'INPI	9914882

DB 540 W /260899

Vos références pour ce dossier : <i>(facultatif)</i>		FR 3863 PG
6 MANDATAIRE		
Nom		GUILLERM
Prénom		Patrice
Cabinet ou Société		BULL S.A.
N °de pouvoir permanent et/ou de lien contractuel		PG 4972
Adresse	Rue	68, route de Versailles
	Code postal et ville	78430 LOUVECIENNES
N° de téléphone <i>(facultatif)</i>		01.39.66.67.28
N° de télécopie <i>(facultatif)</i>		01.39.66.71.71
Adresse électronique <i>(facultatif)</i>		patrice.guillerm@bull.net
7 INVENTEUR (S)		
Les inventeurs sont les demandeurs		<input type="checkbox"/> Oui <input checked="" type="checkbox"/> Non Dans ce cas fournir une désignation d'inventeur(s) séparée
8 RAPPORT DE RECHERCHE		
Établissement immédiat ou établissement différé		<input checked="" type="checkbox"/> <input type="checkbox"/>
Paiement échelonné de la redevance		Paiement en trois versements, uniquement pour les personnes physiques <input type="checkbox"/> Oui <input type="checkbox"/> Non
9 RÉDUCTION DU TAUX DES REDEVANCES		Uniquement pour les personnes physiques <input type="checkbox"/> Requise pour la première fois pour cette invention (<i>joindre un avis de non-imposition</i>) <input type="checkbox"/> Requise antérieurement à ce dépôt (<i>joindre une copie de la décision d'admission pour cette invention ou indiquer sa référence</i>):
Si vous avez utilisé l'imprimé « Suite », indiquez le nombre de pages jointes		
10 SIGNATURE DU DEMANDEUR OU DU MANDATAIRE (Nom et qualité du signataire)		VISA DE LA PRÉFECTURE OU DE L'INPI
 Patrice GUILLERM (Salarié Bull S.A)		

DÉPARTEMENT DES BREVETS

26 bis, rue de Saint Pétersbourg
75800 Paris Cedex 08
Téléphone : 01 53 04 53 04 Télécopie : 01 42 94 86 54

DÉSIGNATION D'INVENTEUR(S) Page N° ... 1 . 1.

(Si le demandeur n'est pas l'inventeur ou l'unique inventeur)

Cet imprimé est à remplir lisiblement à l'encre noire

DB 113 W /260899

Vos références pour ce dossier (facultatif)	FR 3863 PG
N° D'ENREGISTREMENT NATIONAL	99148821
TITRE DE L'INVENTION (200 caractères ou espaces maximum)	
Procédé de création de fichiers de configuration d'objets inclus dans un système informatique.	
E(S) DEMANDEUR(S) :	
BULL S.A. 68, route de Versailles 78430 LOUVECIENNES	
DESIGNE(NT) EN TANT QU'INVENTEUR(S) : (Indiquez en haut à droite «Page N° 1/1» Si il y a plus de trois inventeurs, utilisez un formulaire identique et numérotez chaque page en indiquant le nombre total de pages).	
Nom Andrei	
Prénoms Pierre	
Adresse	Rue 10, rue Hérold
	Code postal et ville 75001 Paris
Société d'appartenance (facultatif)	
Nom Bui-Xuan	
Prénoms Hoan	
Adresse	Rue 1, avenue Gambetta
	Code postal et ville 75020 Paris
Société d'appartenance (facultatif)	
Nom	
Prénoms	
Adresse	Rue
	Code postal et ville
Société d'appartenance (facultatif)	
DATE ET SIGNATURE(S) DU (DES) DEMANDEUR(S) OU DU MANDATAIRE (Nom et qualité du signataire)	
Louvenciennes, le 24 novembre 1999	
	
Patrice GUILLERM (Salarié Bull S.A.)	

DOCUMENT COMPORTANT DES MODIFICATIONS

Un changement apporté à la rédaction des revendications d'origine, sauf si celui-ci découle des dispositions de l'article R.612-36 du code de la Propriété Intellectuelle, est signalé par la mention «R.M.» (revendications modifiées).

Procédé de création de fichiers de configuration d'objets inclus dans un système informatique.

DESCRIPTION

5

Domaine technique

La présente invention se rapporte à un procédé de création de fichiers de configuration d'objets appartenant à un système informatique.

Le système informatique comprend des objets matériels (machines, ...),
10 et/ou logiciels (applications, ...). Ce système est indifféremment un système distribué ou non, hétérogène ou non.

Les objets comprennent des paramètres inclus dans un fichier de configuration. L'invention s'applique à tout fichier de configuration écrit dans un langage de balisage extensible, c'est-à-dire un langage qui présente de
15 l'information encadrée par des balises. Le fichier de configuration est écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer. Un métalangage se définit généralement comme étant un langage utilisé pour décrire un autre langage. Le langage de balisage XML (eXtensible Markup Language), connu de l'homme du métier, est
20 bien adapté à la mise en œuvre de la présente solution. Rappelons que la spécification du langage XML est définie par le consortium W3C (World Wide Web Consortium). Ce consortium est un Organisme de promotion du «World Wide Web», qui met au point des normes et des protocoles ouverts et libres, dans un souci d'Interopérabilité maximale. Le fichier de configuration XML a
25 une structure déclarée dans un fichier de description. Ce fichier de description comprend la description des paramètres de configuration d'un objet et se nomme généralement Définition de Type de Document (DTD). Cette définition

s'effectue selon un formalisme particulier également défini dans la spécification XML du consortium W3C.

L'art antérieur

5 Par définition, l'écriture d'un fichier de configuration dans un langage XML doit obéir à certaines contraintes syntaxiques. En effet, un document XML a une structure logique. Il se compose de descriptions, d'éléments, de commentaires, d'appels de caractère et d'instructions de traitement, qui sont indiqués dans le document par l'intermédiaire d'un balisage explicite. Les
10 éléments sont encadrées par de balises ouvrantes, par exemple <préface>, et des balises fermantes, par exemple </préface>. Les éléments sont structurés et décrivent les paramètres des objets. Les paramètres, comme un attribut d'un objet, peuvent être inclus à l'intérieur même des balises. Par exemple, on peut écrire <LIVRE sujet = k> signifiant que l'attribut sujet de l'élément LIVRE a la
15 valeur K.

Dans notre exemple de réalisation, Les paramètres des objets sont définis dans un fichier de configuration écrit dans un langage XML tel que défini précédemment.

Le problème principal est que les objets à configurer se comptent en
20 milliers et que plusieurs de ces objets peuvent se reposer sur un même fichier de description DTD et avoir des valeurs identiques pour tout ou partie de leurs paramètres. Pour construire le fichier de configuration, l'administrateur du système de gestion doit alors valoriser les paramètres décrit dans le fichier de description autant de fois qu'il y a d'objets se reposant sur ce fichier DTD. Plus
25 précisément, supposons que deux objets B1 et B2 se reposent sur un même fichier de description (DTD). Pour configurer l'objet B1, l'utilisateur doit valoriser tous les paramètres décrit dans le fichier DTD. Pour configurer l'objet B2, l'utilisateur doit à nouveau valoriser tous les paramètres décrit dans le fichier DTD. En conséquence, les paramètres de même valeur sont écrit autant
30 de fois qu'il y a d'objets se reposant sur un même fichier de description.

L'écriture d'un fichier de configuration présente donc des redondances. Le langage XML étant un langage de configuration de bas niveau, un utilisateur est donc contraint d'écrire dans le fichier de configuration des descriptions de paramètres répétitives dont la sémantique est, dans certains cas, sans rapport avec ses besoins, ce qui nécessite une culture importante de sa part de l'ensemble des syntaxes offertes par le langage XML. De ce fait, le fournisseur doit fournir avec le fichier de description une documentation précise. Il est clair qu'un fichier de configuration impose un coût en temps important en écriture.

Un autre problème, lié au nombre important d'objets à décrire, est que si un utilisateur situé sur une machine quelconque du réseau souhaite visualiser des ressources du système informatique configurées dans le fichier de configuration, le système de gestion doit alors transmettre le fichier de configuration à la machine distante par l'intermédiaire du réseau. Lorsque le fichier de configuration a un gros volume, le flux de données entre le système de gestion et l'application cliente peut s'avérer très important et conduire à saturer le système de communication entre les applications clientes et le système de gestion.

Enfin, un autre problème est que la syntaxe définie selon les recommandations du consortium W3C doit être respectée à la lettre tout au long de l'écriture du fichier de configuration. Le risque d'erreur lors de l'écriture d'un fichier de configuration est donc permanent pour un administrateur du système de gestion.

Sommaire de l'invention

Un premier but de la solution est donc de simplifier considérablement l'écriture des fichiers de configuration réduisant en conséquence à la fois le coût en temps d'écriture de celui-ci et le risque d'erreurs d'écriture.

Un deuxième but visé est de réduire la taille du fichier de configuration.

A cet effet, la solution a pour objet un procédé de création d'au moins un fichier de configuration d'objets matériels et/ou logiciels présents dans un

système informatique, ledit fichier de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres desdits objets et se reposant sur un fichier de description 5 définissant des contraintes à respecter sur la structure et la syntaxe lors de l'écriture dudit fichier de configuration, caractérisé en ce qu'il consiste à étendre le fichier de description par au moins un modèle comprenant au moins un paramètre décrit dans le fichier de description, et en ce qu'il consiste à valoriser tout ou partie des paramètres de ce modèle.

10 Il en résulte également un fichier de configuration d'objets matériels et/ou logiciels présents dans un système informatique, ledit fichier de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres desdits objets et se 15 reposant sur un fichier de description définissant des contraintes à respecter sur la structure et la syntaxe lors de l'écriture dudit fichier de configuration, caractérisé en ce que le fichier de description est étendu, en ce que l'extension comprend au moins un modèle incluant au moins un paramètre inclus dans le fichier de description, et en ce que une partie des paramètres de ce modèle 20 sont valorisés.

L'invention sera mieux comprise à la lecture de la description qui suit, donnée à titre d'exemple et faite en référence aux dessins annexés.

25 Description d'un exemple de réalisation

Dans les dessins:

- la figure 1 est une vue synoptique de l'architecture d'un système informatique sur lequel peut s'appliquer la solution,
- la figure 2, est une vue d'un modèle conforme à la présente solution.

Sur la figure 1, on a représenté un système informatique SYS distribué illustrant un exemple de réalisation préféré de la solution. Dans l'exemple illustré, ce système SYS inclut un système de gestion SG et au moins une machine M1. Le système de gestion SG comprend au moins un système d'exploitation, au moins une mémoire de stockage d'informations et au moins un processeur contrôlant le processus du traitement de l'information. Le terme gestion est utilisé pour être conforme à la traduction Afnor (Association Française de NORmalisation) de « Management ». Un système de gestion de machines de type « Open Master » (marque déposée par la société BULL S.A.), connu de l'homme du métier, est particulièrement bien adapté pour la mise en œuvre de la solution. Ce système de gestion peut être assimilé à un ensemble de services qui interagissent entre eux pour donner une représentation objet du monde réel constitué notamment par les machines du système informatique. C'est une représentation objet qu'un administrateur manipule (surveillance, action) pour gérer le monde réel. La représentation objet porte sur des objets virtuels du monde réel et constitue un modèle objet. En d'autres mots, un objet géré par le système de gestion est une vue abstraite, définie pour les besoins de gestion, d'une ressource logique ou physique du système informatique. (disque, processeur, mémoire, etc.) et/ou logiques (fichiers, processus, sémaphores, etc.).

Le système de gestion et les machines qu'il gère constituent une architecture Client/Serveur. Dans une telle architecture, un application cliente interroge le système de gestion pour connaître l'état des objets gérés par le système de gestion. Le mode client/Serveur a l'avantage de permettre à un utilisateur appelé client (ou application cliente) situé sur une machine, par exemple par l'intermédiaire d'un simple micro-ordinateur ou d'une station de travail, de confier une partie de sa tâche ou de ses opérations à effectuer au serveur à savoir le système de gestion. De cette manière, le client dispose d'une capacité de calcul beaucoup plus importante que celle de son micro-ordinateur.

Le système informatique peut être hétérogène. Afin de masquer l'hétérogénéité du système informatique, le système de gestion SG et les machines gérées par le système de gestion comprennent au moins un agent respectif associé à un protocole de gestion. Un agent assure, entre autres, une conversion de protocole.

Le système de gestion est relié à une machine gérée par l'intermédiaire d'un réseau quelconque. Le réseau peut être de type LAN (Local Area Network), WAN (Wide Area Network). Un ensemble de couches logicielles s'interpose entre le système de gestion SG et le réseau RES et entre le réseau et chaque machine. Pour des raisons de simplification de la description, cet ensemble de couches logicielles n'est pas représenté sur la figure 1.

Chaque objet géré comprend des paramètres définis dans un fichier de configuration de préférence écrit dans un langage de description à balisage comportant une structure et incluant tout ou partie des paramètres (nom, au moins un attribut, au moins une action, etc.) des objets. Le fichier de configuration se base sur un fichier distinct appelé fichier de description définissant les contraintes sur la structure et les contraintes syntaxiques des paramètres pour l'écriture dudit fichier de configuration associé à un objet d'une machine. De préférence, le fichier de configuration est écrit dans un langage de type XML, et le fichier de description est un fichier de description de type DTD, connus de l'homme du métier. Dans notre exemple de réalisation, ce fichier de configuration XML et le fichier de description DTD sont inclus dans le système de gestion SG.

Dans notre exemple de réalisation, le fichier de description DTD est centralisé sur le système de gestion de façon à être utilisable par toutes les machines du réseau. De préférence, les objets gérés peuvent être représentés par un arbre, chaque noeud de l'arbre représentant un objet géré. Une application de visualisation APV incluse sur la machine M1 peut interroger le fichier de configuration dans le but de recevoir les paramètres des objets configurés et de visualiser ces objets sur cette machine. De préférence, pour la visualisation des paramètres de configuration des objets, la machine M1 est

munie d'un navigateur standard dit « Internet Explorer », connu de l'homme du métier, dans lequel un programme en langage JAVA est exécuté pour lire le fichier de configuration, accédant ainsi à son contenu et à sa structure, et pour transmettre les informations lues à (aux) applications de visualisation APV.

5

Description de paramètres de configuration d'un objet :

Un objet comprend comme paramètre au moins un attribut. Par exemple, un attribut ID est l'identificateur de l'objet, un autre attribut TYPE désigne son type, et un autre attribut OWNER. De plus, un objet a des propriétés. Dans 10 notre exemple, un objet comprend également comme paramètres:

- le nom de l'objet,
- les actions que l'on peut exécuter sur cet objet incluant l'action ouvrir pour l'ouverture du noeud, l'action fermer pour la fermeture du noeud, l'action développer pour visualiser les noeuds subordonnés à un noeud,
- 15 ■ et des propriétés graphiques de ce noeud incluant le type de police de caractère, l'adresse de l'icône qui lui est associé, et la couleur de fond désirée

Ainsi, dans ce fichier de description DTD, un élément « noeud » est associé à un noeud, et des éléments lui sont subordonnés et sont associés 20 respectivement

- au nom de l'objet
- aux actions (ouvrir, fermer, développer) que l'on peut exécuter sur cet objet,
- et aux propriétés graphiques (police de caractère, icône, couleur de 25 fond) de cet objet.

Des attributs peuvent être associé à un élément. Dans notre exemple de réalisation, l'élément « noeud » comprend trois attributs à savoir :

- un attribut ID désignant son identificateur
- un attribut TYPE désignant son type
- 30 ■ et un attribut désignant son propriétaire

Un fichier de description DTD définissant un objet de l'arbre peut être écrit de la façon suivante, en respectant le formalisme particulier défini dans la spécification XML du consortium W3C :

```

5      <!ELEMENT noeud (noeudNom, noeudActions,
noeudPropriétéGraphique)>
       <!ELEMENT noeudActions (action*)>
       <!ELEMENT noeudNom (groupeNom, adresseNom, versionNom)>
       <!ATTLIST noeud Id CDATA #REQUIRED
10     Type CDATA #REQUIRED
          Owner CDATA #REQUIRED
         >
         <!ELEMENT noeudPropriétégaphique (police, icône, couleur)>
```

15 Dans ce fichier, CDATA #REQUIRED signifie que l'attribut en question doit être un bloc de texte contenant des caractères. De plus, l'écriture « action* » signifie que les actions n'ont pas d'attributs et que la syntaxe de ces actions à utiliser lors de l'écriture du fichier de configuration est du texte.

20 L'écriture <!ELEMENT noeudNom (groupeNom, adresseNom, versionNom)> indique que l'élément « noeudNom » comprend trois éléments (groupeNom, adresseNom, versionNom) qui lui sont subordonné. L'élément « groupeNom » désigne le nom de l'objet, l'élément « adresseNom » désigne l'adresse de l'objet, et l'élément « versionNom » désigne la version de l'objet.

25 Ce fichier de description correspondra dans la suite de la description au fichier de description initialement créé.

Notons que le nombre de paramètres dans notre exemple de réalisation est réduit pour des raisons de simplification de la description. En général un fichier de description comprend un nombre de paramètres plus important.

30 Dans l'exemple illustré, supposons par exemple que trois objets (OBJ1, OBJ2 et OBJ3) se reposent sur le même fichier de description DTD défini plus

haut. Le problème principal est que, pour construire le fichier de configuration, l'administrateur du système de gestion doit valoriser les paramètres décrit dans le fichier de description autant de fois qu'il y a d'objets se reposant sur ce fichier DTD. L'administrateur doit donc compléter le fichier de configuration 5 trois fois. Le coût en temps d'une telle écriture est donc considérable.

A cet effet, la solution consiste à étendre le fichier de description par au moins un modèle comprenant au moins un paramètre inclus dans le fichier de description, et en ce qu'il consiste à valoriser une partie des paramètres de ce modèle d'élément. En L'espèce, la solution consiste à introduire un modèle 10 d'élément dont l'écriture respectent des propriétés définies dans ce qui suit.

Dans notre exemple de réalisation, nous avons à définir un ensemble d'objets dont les seuls paramètres variant entre eux sont

- l'élément « Nom » correspondant au nom des objets (OBJ1, OBJ2 et 15 OBJ3), respectivement (JAZZ, POP, SOL),

- et l'attribut « ID » correspondant à l'identificateur des objets (OBJ1, OBJ2 et OBJ3), respectivement (123, 142, 162).

Ces deux paramètres seront dits indéfinis dans le suite de la 20 description. Dans l'exemple de réalisation, les objets (OBJ1, OBJ2 et OBJ3) ont un nom respectif (JAZZ, POP, SOL) et un identificateur respectif (123, 142, 162).

Conformément à notre hypothèse de départ, les autres paramètres ont la même valeur pour chaque objet (OBJ1, OBJ2 et OBJ3). Ils seront dits définis. Ainsi,

■ la valeur de l'élément correspondant aux actions que l'on peut 25 exécuter est la même pour chaque objet (OBJ1, OBJ2 et OBJ3),

■ la valeur de l'élément correspondant aux propriétés graphiques est la même pour chaque objet (OBJ1, OBJ2 et OBJ3) ,

et la valeur des attributs

- TYPE désignant le type du objet est le même pour chaque objet (OBJ1, OBJ2 et OBJ3),
- et OWNER désignant son propriétaire est le même pour chaque objet (OBJ1, OBJ2 et OBJ3).

5 Pour des raisons de simplification de la description, on donnera des valeurs arbitraires aux paramètres définis. Dans notre exemple, la valeur de l'élément correspondant aux actions que l'on peut exécuter sur chaque objet (OBJ1, OBJ2 et OBJ3) est ACT1 pour la commande « ouvrir », ACT2 pour la commande « fermer », et ACT3 pour la commande « développer ». De même,

10 10 la valeur de l'élément correspondant aux propriétés graphiques de chaque objet (OBJ1, OBJ2 et OBJ3) est PRO1 pour la police de caractère, PRO2 pour l'icône, et PRO3 pour la couleur de fond. Enfin, les attributs TYPE et OWNER prennent pour chaque objet (OBJ1, OBJ2 et OBJ3) respectivement les valeurs « snmp » et « opérateur ».

15 Les deux étapes principales du procédé conforme à la solution sont respectivement

- l'écriture du fichier de description DTD et de l'extension associé à ce fichier constitué par au moins un modèle d'élément (étape 1),
- et l'écriture du fichier de configuration résultant de la valorisation des 20 paramètres indéfinis du modèle d'élément (étape 2).

Ecriture du fichier de description et introduction de la notion de modèle d'élément dans un fichier de description :

La première étape doit réalisée en respectant certaines propriétés. Les paramètres de ces objets dont la valeur est invariante étant identifiés, la 25 solution consiste à introduire le modèle d'élément MODELE dans le fichier de description DTD créé. Le modèle d'élément se distingue des autres éléments du fichier de description en ce sens qu'il comporte au moins un paramètre avec une valeur.

Premièrement, le modèle comprend, en respectant le formalisme d'écriture d'un fichier de description DTD défini dans la spécification XML du consortium W3C,

- une entête MODELE avec un nom spécifique « tnoeud »
- 5 - et une référence à un élément « noeud » défini du fichier de description DTD créé initialement.

Le modèle MODELE peut être écrit de la façon suivante :

```
<!MODELE tnoeud ELEMENT =noeud >
```

signifiant que le modèle MODELE à un nom spécifique « tnoeud » et
10 qu'il se base sur la description de l'élément « noeud » défini au préalable dans le fichier de description (DTD).

Deuxièmement, dans ce modèle, l'écriture des éléments indéfinis est particulière. De façon à distinguer un élément défini d'un élément indéfini dans le modèle d'élément, les éléments à définir sont repérés dans ce modèle par
15 l'intermédiaire d'une balise spécifique dont l'entête est par exemple <!DEFINIR...>. De plus, les éléments à définir sont identifiés par un nom « tnoeudNom » et une référence à un élément noeudNom du fichier de description initial précisant sur quel élément du fichier de description préalablement défini se base le modèle d'élément « tnoeudNom ». Dans notre
20 exemple, un élément à définir peut s'écrire de la façon suivante :

```
<!DEFINIR tnoeudNom...ELEMENT noeudNom >.
```

A la différence des éléments indéfinis, les éléments définis du modèle d'élément sont écrits de la même façon que pour l'écriture d'un fichier de
25 configuration XML.

Troisièmement, l'écriture des attributs indéfinis respecte des propriétés. Dans ce modèle, les attributs à définir et définis, la solution consiste à introduire deux mots clés DEFINIR et DEFINI indiquant qu'un paramètre

attribut est à définir (DEFINIR) ou est défini (DEFINI). Dans notre exemple, l'attribut ID est à définir lors de l'écriture du fichier de configuration, tandis que les attributs TYPE et OWNER sont définis et prennent respectivement les valeurs « snmp » et « opérateur ». Dans notre exemple, la liste d'attributs est relative au modèle d'élément MODELE « noeud » et peut s'écrire de la façon suivante :

```
10 <!ATTLIST noeud
      ID DEFINIR
      TYPE DEFINI « snmp »
      OWNER DEFINI « opérateur »
      >
```

En définitive, le fichier de description DTD qui découle d'une telle configuration peut s'écrire de la façon suivante :

```
15 <!MODELE noeud ELEMENT=noeud
      <!DEFINIR noeudNom ELEMENT=noeudNom>
      < noeudActions >
          <action nom=ouvrir> ACT1</action>
          <action nom=fermer>ACT2</action>
          <action nom=développer>ACT3</action>
      20 </ noeudActions >
      < noeudPropriétésgraphiques >
          <police de caractères PRO1... />
          <Icône>PRO2</Icône>
          < couleur de fond PRO3/>
      25 < / Propriétésgraphiques >
      >
      <!ATTLIST noeud
          ID DEFINIR
          TYPE DEFINI « snmp »
```

OWNER DEFINI « opérateur »

>.

Ce fichier constitue un facteur commun FC pour l'écriture du fichier de configuration et décrit les paramètres (élément et/ou attribut) à définir.

5

Ecriture du fichier de configuration

La figure 2 est une vue du fichier de configuration qui résulte de l'utilisation du modèle d'élément.

L'écriture du fichier de configuration correspond à la deuxième étape.

- 10 Cette opération consiste à utiliser le modèle d'élément MODELE défini dans le fichier de description DTD et à valoriser les éléments et attributs indéfinis. Lors de l'écriture du fichier de configuration, la solution consiste à utiliser la partie comprenant les paramètres valorisés comme facteur commun, et en ce que l'écriture se limite à la valorisation des paramètres ne comportant pas de
- 15 valeur.

En l'espèce, trois objets (OBJ1, OBJ2 et OBJ3) ayant pour nom respectif (JAZZ, POP, SOL) et un identificateur respectif (123, 142, 162) sont à configurer et l'écriture du fichier de configuration pour ces trois objets se limite à écrire les trois blocs suivants (B1, B2 et B3) :

- 20 (B1)

```

<t noeud Id="«123»>
  <t noeudNom>
    <noeudNom>
      <groupeNom> JAZZ </groupeNom>
      <adresseNom> db0 </ adresseNom >
      <versionNom> 8.0 <versionNom>
    </noeudNom>

```

```

</tnoeudNom>
</tnoeud>

(B2)

5 <tnoeud Id="«142»">
  <tnoeudNom>
    <noeudNom>
      <groupeNom> POP </groupeNom>
      <adresseNom> db1 </adresseNom>
    10 <versionNom> 8.1 </versionNom>
    </noeudNom>
    </tnoeudNom>
  </tnoeud>

15 (B3)
<tnoeud Id="«162»">
  <tnoeudNom>
    <noeudNom>
      <groupeNom> SOL </groupeNom>
    20 <adresseNom> db2 </adresseNom>
      <versionNom> 8.2 </versionNom>
    </noeudNom>
    </tnoeudNom>
  </tnoeud>

25 Le nombre d'invocation du modèle d'élément correspond au nombre
d'objets se reposant sur ce modèle MODELE. Ces trois blocs ont pour même
facteur commun celui créé dans le fichier de description.

Selon une variante, un modèle d'élément peut contenir d'autres modèles
d'éléments. Ainsi, dans un fichier de configuration, on peut utiliser à l'intérieur

```

d'une référence de modèle d'élément (par exemple « tnoeud ») une autre référence de modèle d'éléments. En effet, supposons qu'il existe dans le fichier de description DTD un modèle d'élément « tOracleNoeudNom », dont les éléments définis sont

- 5 ■ l'élément nommé groupeNom
- et l'élément versionNom

et dont l'élément indéfini est adresseNom. L'écriture de ce modèle d'élément peut être le suivant :

10 <!MODELE tOracle8NodeName ELEMENT=noeudNom
 <groupeNom> Oracle </groupeNom>
 <!DEFINIR tOracleDb ELEMENT=adresseNom>
 <versionNom> 8.0 <versionNom>
 >

15 De cette façon, lors de l'écriture du fichier de configuration, on peut utiliser le modèle « tOracleNoeudNom » pour définir l'élément noeudNom. Le fichier de configuration s'écrit alors de la façon suivante :

<tnoeud id="123">
 20 <tnoeudNom>
 <tOracle8NoeudNom>
 <tOracleDb>
 <adresseNom> db0 </adresseNom>
 </tOracleDb>
 25 </tOracle8NoeudNom>
 </tnoeudNom>
 </tnoeud>

D'une manière générale, la solution a pour objet un procédé de création,
 30 dans un système informatique, d'au moins un fichier de configuration d'au moins un objet matériel et/ou logiciel comprenant des paramètres, ledit fichier

de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres dudit objet et se reposant sur un fichier de description définissant des contraintes à respecter sur sa structure et sa syntaxe lors de son écriture, caractérisé en ce qu'il consiste, avant l'écriture du fichier de configuration, à étendre le fichier de description par au moins un modèle comprenant au moins un paramètre décrit dans le fichier de description, et à valoriser tout ou partie des paramètres de ce modèle.

10 Ensuite, lors de l'écriture du fichier de configuration, la solution consiste à utiliser la partie du modèle comprenant les paramètres valorisés comme facteur commun, et en ce que l'écriture du fichier de configuration se limite à la valorisation des paramètres ne comportant pas de valeur.

15 Lors de la création du modèle, on a vu que la solution peut consister tout d'abord à regrouper les objets se reposant sur le même fichier de description, ensuite à identifier les paramètres dont la valeur est identique entre tous ces objets, et enfin à valoriser ces paramètres pour constituer un facteur commun dans ce modèle.

20 La solution consiste par exemple, lors de l'écriture du fichier de configuration, si au moins deux objets se reposent sur le même modèle, à utiliser le facteur commun et à valoriser uniquement le restant des paramètres de ce modèle autant de fois qu'il y a d'objets se reposant sur ce modèle d'élément.

25 Le langage utilisé est extensible. Dans notre exemple, on a vu que la solution consiste à donner un nom pour identifier le modèle dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence du fichier de description, cette référence définissant les contraintes à respecter sur la structure et la syntaxe de ce modèle. On a vu également que dans notre exemple, la solution consiste à introduire dans un modèle deux mots clés

DEFINIR et DEFINI indiquant qu'un paramètre est à définir (DEFINIR) ou est défini (DEFINI) dans ce modèle.

De préférence, le langage du fichier de configuration est le langage XML, la solution consistant à prendre comme paramètre un élément et/ou un attribut d'un objet. Selon cette variante, la solution consiste à étendre le fichier de description par au moins un modèle d'élément comprenant au moins un paramètre (élément et/ou attribut) décrit dans le fichier de description, et à valoriser tout ou partie des paramètres de ce modèle d'élément. De même, la solution consiste à donner un nom pour identifier le modèle d'élément dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence à un élément du fichier de description, cette référence définissant les contraintes à respecter sur la structure et la syntaxe de ce modèle.

Enfin, on a vu que, sur requête d'une application utilisant le fichier de configuration, la solution peut consister à transmettre le facteur commun et les blocs résultant de la valorisation des éléments indéfinis.

Enfin, il en résulte un fichier de configuration d'au moins un objet matériel et/ou logiciel comprenant des paramètres, ledit fichier de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres dudit objet et se reposant sur un fichier de description définissant des contraintes à respecter sur sa structure et sa syntaxe lors de son écriture, caractérisé en ce que le fichier de description est étendu, et en ce que l'extension comprend au moins un modèle.

En conclusion, la solution offre de nombreux avantages. Un premier avantage est la simplification considérable de l'écriture d'un fichier de configuration. En effet, les paramètres définis étant valorisés dans le modèle d'élément, l'écriture du fichier de configuration se limite à la valorisation des paramètres indéfinis. En conséquence, l'administrateur évite ainsi la répétition de paramètres identiques entre objets. Le coût en temps d'écriture et le risque d'erreurs d'écriture lors de l'écriture du fichier de configuration est largement réduit. De plus, l'introduction d'un facteur commun dans le modèle réduit

considérablement la taille du fichier de configuration. Avantageusement, sur requête d'une application cliente, le système de gestion transmet le fichier de configuration incluant le facteur commun du modèle d'élément et, pour chacun des objets, les paramètres de ce modèle d'élément qui ont été valorisés lors de 5 l'écriture du fichier de configuration. La taille du fichier de configuration étant réduit, le transfert de ce fichier s'effectue donc plus rapidement.

REVENDICATIONS

- 1- Procédé de création, dans un système informatique, d'au moins un fichier de configuration d'au moins un objet matériel et/ou logiciel comprenant des paramètres, ledit fichier de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres dudit objet et se reposant sur un fichier de description définissant des contraintes à respecter sur sa structure et sa syntaxe lors de son écriture, caractérisé en ce qu'il consiste, avant l'écriture du fichier de configuration,
 - à étendre le fichier de description par au moins un modèle comprenant au moins un paramètre décrit dans le fichier de description,
 - et à valoriser tout ou partie des paramètres de ce modèle.
- 2- Procédé selon la revendication 1, caractérisé en ce qu'il consiste, lors de l'écriture du fichier de configuration, à utiliser la partie du modèle comprenant les paramètres valorisés comme facteur commun, et en ce que l'écriture du fichier de configuration se limite à la valorisation des paramètres ne comportant pas de valeur.
- 3- Procédé selon la revendication 1 ou 2, caractérisé en ce qu'il consiste, lors de la création du modèle, à regrouper les objets se reposant sur le même fichier de description et ensuite à identifier les paramètres dont la valeur est identique entre tous ces objets, et en ce qu'il consiste à valoriser ces paramètres pour constituer un facteur commun dans ce modèle.
- 4- Procédé selon l'une des revendications 1 à 3, caractérisé en ce qu'il consiste, lors de l'écriture du fichier de configuration, si au moins deux objets se reposent sur le même modèle, à utiliser le facteur commun et à valoriser uniquement le restant des paramètres de ce modèle autant de fois qu'il y a d'objets se reposant sur ce modèle d'élément.

5- Procédé selon l'une des revendications 1 à 4, caractérisé en ce que le langage est extensible et en ce qu'il consiste à donner un nom pour identifier le modèle dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence du fichier de description, cette référence définissant les 5 contraintes à respecter sur la structure et la syntaxe de ce modèle.

6- Procédé selon l'une des revendications 1 à 5, caractérisé en ce que le langage est extensible et en ce qu'il consiste à introduire dans un modèle deux mots clés DEFINIR et DEFINI indiquant qu'un paramètre est à définir (DEFINIR) ou est défini (DEFINI) dans ce modèle.

10 7- Procédé selon l'une des revendications 1 à 6, caractérisé en ce que le langage est le langage XML, et en ce qu'il consiste à prendre comme paramètre un élément et/ou un attribut d'un objet.

15 8- Procédé selon la revendication 7, caractérisé en ce qu'il consiste à étendre le fichier de description par au moins un modèle d'élément comprenant au moins un paramètre (élément et/ou attribut) décrit dans le fichier de description, et à valoriser tout ou partie des paramètres de ce modèle

20 9- Procédé selon la revendication 7 ou 8, caractérisé en ce qu'il consiste à donner un nom pour identifier le modèle d'élément dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence à un élément du fichier de description, cette référence définissant les contraintes à respecter sur la structure et la syntaxe de ce modèle.

10- Procédé selon l'une des revendications 7 à 9, caractérisé en ce qu'il consiste à inclure dans un modèle d'élément au moins un modèle d'élément.

25 11- Procédé selon l'une des revendications 7 à 10, caractérisé en ce qu'il consiste, sur requête d'une application utilisant le fichier de configuration, à transmettre le facteur commun et les blocs résultant de la valorisation des éléments indéfinis.

30 12- Fichier de configuration d'au moins un objet matériel et/ou logiciel comprenant des paramètres, ledit fichier de configuration étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des

paramètres dudit objet et se reposant sur un fichier de description définissant des contraintes à respecter sur sa structure et sa syntaxe lors de son écriture, caractérisé en ce que le fichier de description est étendu, et en ce que l'extension comprend au moins un modèle tel que défini dans l'une des
5 revendications 1 à 11.

feuille rectifiée

REVENDICATIONS

- 1- Procédé de création, dans un système informatique, d'au moins un fichier de configuration d'au moins un objet matériel et/ou logiciel comprenant des paramètres, ledit fichier de configuration étant stocké dans une mémoire de stockage d'information et étant écrit en utilisant un métalangage de description dont le format est indépendant du matériel et/ou logiciel à configurer, ce fichier de configuration incluant tout ou partie des paramètres dudit objet et se reposant sur un fichier de description définissant des contraintes à respecter sur sa structure et sa syntaxe lors de son écriture, caractérisé en ce qu'il consiste, avant l'écriture du fichier de configuration,
 - à étendre le fichier de description par au moins un modèle comprenant au moins un paramètre décrit dans le fichier de description,
 - et à valoriser tout ou partie des paramètres de ce modèle.
- 15 2- Procédé selon la revendication 1, caractérisé en ce qu'il consiste, lors de l'écriture du fichier de configuration, à utiliser la partie du modèle comprenant les paramètres valorisés comme facteur commun, et en ce que l'écriture du fichier de configuration se limite à la valorisation des paramètres ne comportant pas de valeur.
- 20 3- Procédé selon la revendication 1 ou 2, caractérisé en ce qu'il consiste, lors de la création du modèle, à regrouper les objets se reposant sur le même fichier de description et ensuite à identifier les paramètres dont la valeur est identique entre tous ces objets, et en ce qu'il consiste à valoriser ces paramètres pour constituer un facteur commun dans ce modèle.
- 25 4- Procédé selon l'une des revendications 1 à 3, caractérisé en ce qu'il consiste, lors de l'écriture du fichier de configuration, si au moins deux objets se reposent sur le même modèle, à utiliser le facteur commun et à valoriser uniquement le restant des paramètres de ce modèle autant de fois qu'il y a d'objets se reposant sur ce modèle d'élément.

5- Procédé selon l'une des revendications 1 à 4, caractérisé en ce que le langage est extensible et en ce qu'il consiste à donner un nom pour identifier le modèle dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence du fichier de description, cette référence définissant les contraintes à respecter sur la structure et la syntaxe de ce modèle.

6- Procédé selon l'une des revendications 1 à 5, caractérisé en ce que le langage est extensible et en ce qu'il consiste à introduire dans un modèle deux mots clés DEFINIR et DEFINI indiquant qu'un paramètre est à définir (DEFINIR) ou est défini (DEFINI) dans ce modèle.

10 7- Procédé selon l'une des revendications 1 à 6, caractérisé en ce que le langage est le langage XML, et en ce qu'il consiste à prendre comme paramètre un élément et/ou un attribut d'un objet.

15 8- Procédé selon la revendication 7, caractérisé en ce qu'il consiste à étendre le fichier de description par au moins un modèle d'élément comprenant au moins un paramètre (élément et/ou attribut) décrit dans le fichier de description, et à valoriser tout ou partie des paramètres de ce modèle

9- Procédé selon la revendication 7 ou 8, caractérisé en ce qu'il consiste à donner un nom pour identifier le modèle d'élément dans le fichier de description, et en ce qu'il consiste à inclure dans le modèle une référence à un élément du fichier de description, cette référence définissant les contraintes à respecter sur la structure et la syntaxe de ce modèle.

20 10- Procédé selon l'une des revendications 7 à 9, caractérisé en ce qu'il consiste à inclure dans un modèle d'élément au moins un modèle d'élément.

11- Procédé selon l'une des revendications 7 à 10, caractérisé en ce qu'il consiste, sur requête d'une application utilisant le fichier de configuration, à transmettre le facteur commun et les blocs résultant de la valorisation des éléments indéfinis.

sys

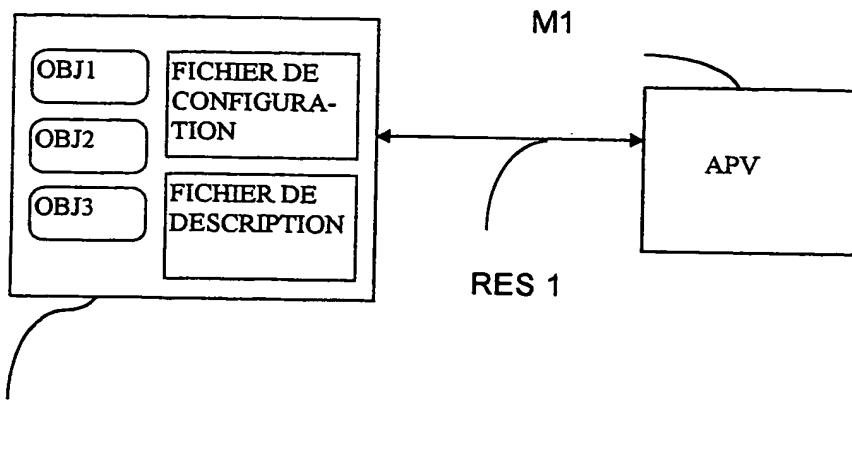


Figure 1

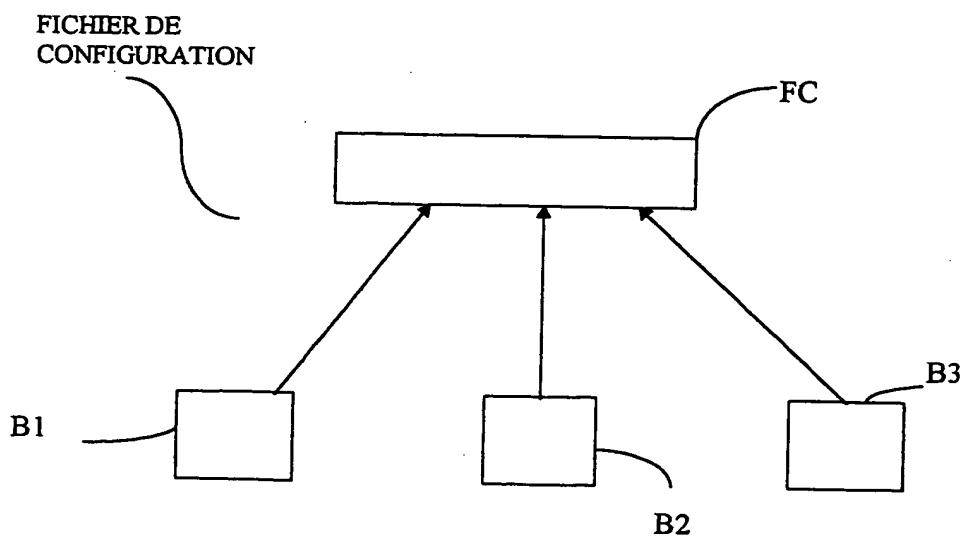


Figure 2